8. List, Tuple en String

List

Een **list variabele** is een lijst elementen. We noteren de list met vierkante haken en gebruiken komma als scheiding tussen de elementen. De elementen kunnen een verschillend type hebben.

```
List1 = [1, 2, 3]
List2 = [‘Jan’, ‘Pieters’, 1980, ‘Lier]
```

We benaderen de elementen met een index. Het eerste element heeft index 0. Python-programmeurs tellen vanaf 0. Andere programmeurs ook.

```
A = List1[1]          # A = 2
B = List2[1:3]        # B = [‘Pieters’, 1980], elementen 1 tot 2
C = List2[2:]         # 2de tot laatste element
List1[1] = 5          # List1 = [1, 5, 3], verander element 1
```

Het sterretje (*) is het herhaalteken. Met het plusteken (+) voegen we twee lijsten samen.

```
List3 = List1 *2      # List3 = [1, 5, 3, 1, 5, 3]
List4 = List1 + List2 # List 4 = [‘Jan,’Pieters’,1980,’Lier’,1,5,3]
```

del verwijdert een element uit een list.

```
del List3[3]          # List3 =[1, 5, 3, 5, 3]
```

Een list kan een element zijn van een andere list. Dat gebruiken wiskundigen om een matrix voor te stellen.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Figuur 34: matrix A

De matrix hier boven noteren we in Python als

```
A = [[1,2,3],[4,5,6],[7,8,9]
```

Tuple

Een **tuple variabele** lijkt heel sterk op een list. Het verschil is dat we na de definitie de elementen niet kunnen veranderen. Een tuple noteren we met ronde haken.

Voorbeelden:

```
Tuple1 = (1, 2, 3)
Tuple2 = ('Jan', 'Pieters', 1980, 'Lier)

A = Tuple1[1]             # A = 2
B = Tuple2[1:3]           # B = ['Pieters', 1980, 'Lier'
C = Tuple2[2:]            # 2de tot laatste element
Tuple1[1] = 5             # FOUT!, elementen kunnen niet wijzigen
Tuple3 = Tuple1 *2        # Tuple3 = (1, 5, 3, 1, 5, 3)
Tuple4 = Tuple1 + Tuple2  # Tuple 4 = ['Jan,'Pieters',1980,4Lier,1,5,3]
```

String

Een **string variabele** is een stuk tekst of een openvolging van letters, cijfer en andere afdrukbare tekens. Python behandels een string als een list. We kunnen afdzonderlijke tekens van die string veranderen.

Voorbeelden:

```
naam = "seele"
a = naam[1]               # a = 'e'
naam[0] = 'S'             # naam = "Seele"
```

Stringbewerkingen

+ voegt strings samen:

```
naam1 = "Herr"
naam2 = "  Seele"
naam3 = naam1 + naam2
resultaat: naam3 = "Herr Seele"
```

***** herhaalt een string:

```
naam4 = naam1 *2
resultaat: naam4 = "HerrHerr"
```

[ ] geeft een karakter uit een string:

```
c = naam1[0]
resultaat: c = "H"
```

[:] geeft een aantal karakters uit een string:

```
c = naam1[1:3]
resultaat: c = "er" (karakters 1 tot 3, 3 niet inbegrepen)
```

in gaat na of een karakter in een string zit:

```
p = ("r" in naam1)
resultaat: p = True
```

not in gaat na of een karakter niet in een string zit:

```
p = ("r" not in naam1)
resultaat: p = False
```

Het opbouwen van een string

In een volgend hoofdstuk gaan we kleine beeldschermen gebruiken. De meeste van die dingen verwachten een string als input. Die string moeten we opbouwen met stukken tekst, getallen, resultaten van berekeningen en resultaten van sensoren. Wie hebben iets gelijkaardigs gedaan bij functie *print* in hoofdstuk *Variabelen in Python*. We werken daar met een string met plaatshouders.

voorbeeld:

```
dag = 14
maand = 2
jaar = 1954
String 1 = "%02u/%02u/%04u" %(dag, maand, jaar)
```

Het %-teken stelt een plaatshouder voor. Letter u in de plaatshouder zegt dat we op die plaats een geheel getal gaan weergeven (=unsigned integer). Met de optionele cijfers voor de u geven we aan met hoeveel cijfers de integers afgebeeld worden. %02u: een geheel getal met 2 cijfers, lege plaatsen vooraan worden 0. De constructie achteraan geeft dan aan welke variabelen/getallen de plaatshouders vervangen.Het resultaat van de constructie wordt dan: String1 = "14/02/1954"

voorbeeld:

```
meting = 3.237
string2 = "temperatuur = %f.2" %(meting)
```

Resultaat: string2 = “temperatuur = 3.24”
(%f.2 geeft een float-getal met 2 cijfers na de komma.

. Mogelijke plaatshouders zijn:

%c	Character
%s	string
%u	unsigned integer
%d	signed integer
%f	float
%x of %X	Hexadecimaal getal
%e of %E	Exponent notation

Het minimaal aantal plaatsen geef je aan met cijfers.

- **%4u**: drukt af als een geheel getal en voorzie 4 plaatsen minimaal. Lege plaatsen worden blanco.
- **%04u**: drukt af als een geheel getal en voorzie 4 plaatsen minimaal. Lege plaatsen worden 0.
- **%06.2f**: drukt af als een kommagetal met twee cijfers na de komma. Voorzie 6 plaatsen minimaal. Lege plaatsen vooraan worden 0.

Enkele niet afdrukbare tekens in een string zijn:

- \n: begin een nieuwe lijn
- \tab: naar volgende tab-plaats
- \a: een belletje